

## Detailed Design – Minted web services client component

### Owners and List of Contacts

Name	Email	Phone	Role
Carol Shergold	carols@sussex.ac.uk		Minted project senior developer

### Revision History

Date	Reason for change(s)	Author(s)
10-Apr-2006	First Draft	Carol Shergold
	Second draft after discussion with Rory and Andy	Carol Shergold

## 1. Summary

The IMS Enterprise 1.1 specification outlines a schema for person data, group data and for membership data. A membership record defines a relationship between person and group. The specification is defined at:

<http://www.imsglobal.org/enterprise/index.html>

There is a further specification that defines the ways in which a web service can create, read, update and delete data in the IMS enterprise format. This is known as the IMS Enterprise Services specification, and is defined at:

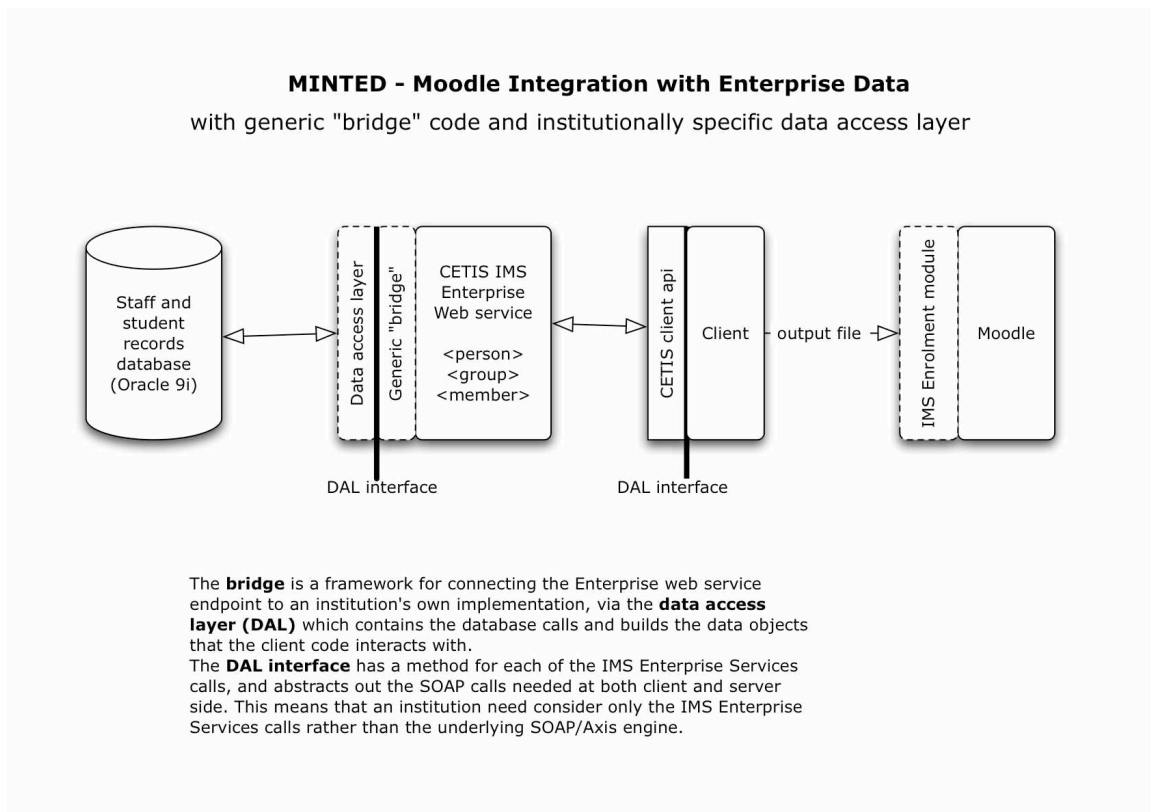
<http://www.imsglobal.org/es/index.html>

In our context, a person can be either a lecturer/course organiser/administrator or a student/learner/participant.

A group is a course running as part of a university degree programme, or a teaching group within that course.

The **cetis-es** toolkit is a java based toolkit for the development of a SOAP webservice that implements the IMS Enterprise Services specification.

The overall system structure for the MINTED project is represented in this diagram:



The particular component from this diagram that this specification relates to is the “web service client”.

The simple case client is a “requester” that calls methods from the Data Access Layer and creates an XML file that can be utilised by the Moodle plugin “IMS Enrolment”.

The core requirement is that the requester can be invoked from the command line, either manually or via a cron job.

```
java -cp <path> requester <webservice endpoint> <course code | null>
```

The webservice endpoint is a mandatory argument that gives the URL for the running web service represented by the box “CETIS IMS Enterprise web service” in the diagram above.

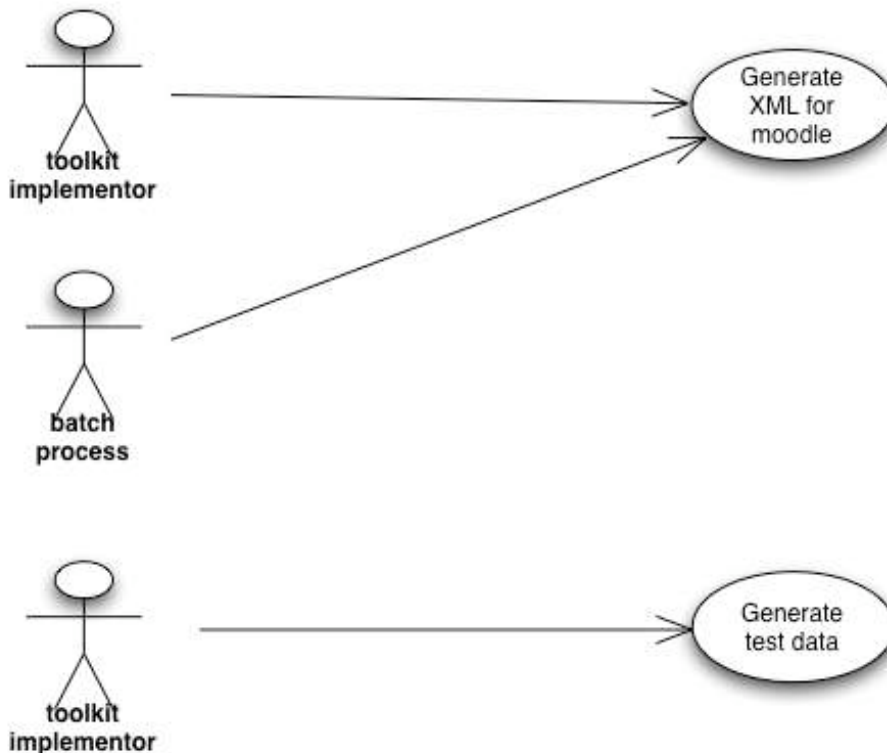
The course code argument is optional. If supplied, then data relating to the specified course is returned. If the argument is omitted, then all data is returned.

The client requester handles all communication with the CETIS ES web service and writes the appropriate XML data format required by the Moodle IMS Enrolment plugin.

A crucial point about the component is that it needs to be very simple for another institution to use it, so we want to abstract out any institutionally specific data into a config file or equivalent.

A useful additional feature of the client code would be that it could be used to demonstrate/test the various methods of the IMS Services specification as implemented in a web service, for example `ReadGroup(IdentifierDType id)`

This is a simple UML use case for the requestor:



## 2. Hardware Requirements

<List all hardware requirements needed to run the system. Include server and client configurations.>

## 3. Software Requirements

The client requester will be written in Java (version 1.4.2 or 1.5) .

The “requester” will use the Data Access Layer and bridge that is a component of the MINTED project, and is the part of the web service that maps between the IMS Enterprise Services calls and the actual database or database API in use here at Sussex.

## 4. Presentation Layer

<This section describes all the screens and reports needed to deliver all the functional requirements. Include screen descriptions, screen shots, report descriptions and report shots. The client should understand that the final product may not be exactly as listed here but the functionality will stay the same. During coding, we may merge or separate screens to achieve a nicer user interface and to promote reusability of components.>

### 4.1 Screens

#### 4.1.1 Screen: request page

A web page GUI for the requester is a highly desirable rather than mandatory option.

It would be useful, as it would provide a convenient focus for any demonstrations of the system that we have to do.

If developed, it would have JISC badging, and an easy way of adding institution and project badging (e.g. in our case, the U of Sussex logo and the Minted project name).

It would display the web service endpoint and allow this to be edited for demo purposes.

It would show the status of that web service, e.g. whether or not it was running.

It would provide a simple html form that could be used to either specify a course code or to request output on all courses.

<screen shot to be added>

A further feature of the web page is that it could allow methods from the data access layer to be called directly, and so provide a more general client for organisations who wish to test their implementations of the web service.

<b>Description</b>	The request page is an optional part of the interface, but would provide a simple way to interact with the web service and would be very useful for giving demos of the system.
<b>Security issues</b>	<List the security groups that may use the screen>

<b>Data</b>	<Describe each field on the screen and how it is derived>
<b>Actions</b>	<Describe each action that can be performed from this screen. This usually involves command buttons, menu items, etc. >

## 4.2 Output

### 4.2.1 IMS Enterprise specification data output file

<b>Description</b>	The output file from the requester is in IMS Enterprise 1.1 specification XML format, as required by the Moodle IMS Enterprise Enrolment plugin.  It would allow for institutional modification.
<b>Security Group</b>	<List the security groups that may use the report >
<b>Export Formats</b>	XML, IMS Enterprise specification 1.1
<b>Frequency</b>	The output will be generated * on demand, e.g. from the command line or via the web page described in section 4.1 above * on a batch basis by a cron job

An example of the required format for this data is given in Appendix 1.

The file location would be stored in an institutional configuration file.

The filename stem would be stored in an institutional configuration file.

## 5. Business Layer

<Define all of the objects necessary to support the presentation layer>

Determine the pathname for the output file:

The file path would be stored in an institutional configuration file.

<Handle issue where more than 1 request is issued before output file has been uploaded into Moodle>

The basic operation of the requester working at course level is:

(a) Find all people who have membership connection with the course using the Data Access Layer method `ReadPersonsForGroup`

(b) Write required data for person schema to output file

(c) Get course data using Data Access Layer method `ReadGroup`

(d) Write required data for course to file

(e) Get membership data Data Access Layer method `ReadGroup`

(f) Write membership data to file

(g) Close connections

## 6. Database Layer

There is no database layer required for this component of the project, as the CETIS-es web service Data Access Layer is carrying out all the interaction with the database.

## 7. Other Design Considerations

### 7.1 Conversion Modules

Describe any programs that are used to import, convert or munge data.

### 7.2 Archive and Purge Modules

Describe any programs that are used to perform special functions like purging and archiving.

### 7.3 Backup and Recovery Design

Describe the process for backing up and recovering lost data.

### 7.4 Security Architecture

Describe how security will be enforced and at what level (or groups). If using NT, address the NT security model, IIS security, Secure Socket Layers, etc. if applicable.

### 7.5 System Interfaces

Describe how we will feed or receive feeds from other systems. This may be via replication, BCP, SQL Loader, etc. Describe the data that will flow between the systems, how often it will be updated, and who the system owners for each system are.

### 7.6 Batch Jobs

Describe any jobs that will run regularly to perform batch based commands.

### 7.7 Performance and Response Time Considerations

Explain how we will design for maximum response time. This includes the use of Stored Procedures, de-normalized data (if applicable), server configuration (size, memory, etc), programming techniques, and the use of database tools such as Oracle's Explain Plan and SQL Server's Show Plan.

### 7.8 Platform Dependence and Installation Considerations

Explain the installation (or setup) process in which the client must use to get the system up and running along with our plan for ensuring it will run on all the platforms requested by the client (Windows 95, 98, NT 4, 5, etc).

### **7.9 Localization Considerations**

Explain our design for handling issues specific to localization (European date and postal code format, etc.), if any.

### **7.10 Other Modules**

Describe any other miscellaneous programs.

## Appendix 1 - Sample output

```
<person recstatus="01">
  <sourcedid>
    <source>wt:ims</source>
    <id>102131</id>
  </sourcedid>
  <userid>melanieg</userid>
  <name>
    <fn><![CDATA[Melanie Green]]></fn>
    <nickname><![CDATA[]]></nickname>
    <n>
      <family><![CDATA[Green]]></family>
      <given><![CDATA[Melanie]]></given>
    </n>
  </name>
  <email>M.J.Green@sussex.ac.uk</email>
</person>
```

```
<person recstatus="01">
  <sourcedid>
    <source>wt:ims</source>
    <id>148903</id>
  </sourcedid>
  <userid>ekk20</userid>
  <name>
    <fn><![CDATA[Eva Kawuma]]></fn>
    <nickname><![CDATA[]]></nickname>
    <n>
      <family><![CDATA[Kawuma]]></family>
      <given><![CDATA[Eva]]></given>
    </n>
  </name>
  <email>ekk20@sussex.ac.uk</email>
</person>
```

```
<group recstatus="01">
  <sourcedid>
    <source>wt:ims</source>
    <id>Q1013:UGYR06/07</id>
  </sourcedid>
  <description>
    <short>Q1013</short>
    <long><![CDATA[Research Project]]></long>
    <full><![CDATA[In the autumn term, you will attend four seminars in which research,
writing, and editing methods are discussed and practised. In the spring and summer terms, you
will work with a supervisor and a peer group in researching your topic and writing up and
presenting the results. This course culminates in the production of an 8,000-word dissertation on
a faculty-approved topic of your choice in linguistics or English language.]]></full>
  </description>
  <org>
```

```
<id><![CDATA[LINGUISTICS & ENGLISH LANGUAGE]]></id>
<orgunit><![CDATA[Linguistics and English Language]]></orgunit>
</org>
<timeframe>
  <begin>2006-10-02</begin>
  <end>2007-06-22</end>
  <adminperiod>UGYR06/07</adminperiod>
</timeframe>
</group>
```

```
<membership recstatus="01">
  <sourcedid>
    <source>wt:ims</source>
    <id>Q1013:UGYR06/07</id>
  </sourcedid>
  <member>
    <sourcedid>
      <source>wt:ims</source>
      <id>102131</id>
    </sourcedid>
    <idtype idtype="1"/>
    <role roletype="05">
      <status>1</status>
    </role>
    <extension>
      <cohort></cohort>
    </extension>
  </member>
</membership>
```

```
<membership recstatus="01">
  <sourcedid>
    <source>wt:ims</source>
    <id>Q1013:UGYR06/07</id>
  </sourcedid>
  <member>
    <sourcedid>
      <source>wt:ims</source>
      <id>148903</id>
    </sourcedid>
    <idtype idtype="1"/>
    <role roletype="01">
      <status>1</status>
    </role>
    <extension>
      <cohort></cohort>
    </extension>
  </member>
</membership>
```